



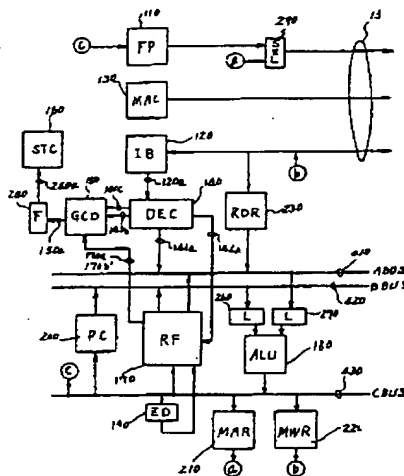
PATENT ABSTRACTS OF JAPAN

(11) Publication number: **01076325 A**(43) Date of publication of application: **22.03.89**(51) Int. Cl. **G06F 9/38**(21) Application number: **62234568**(22) Date of filing: **18.09.87**(71) Applicant: **HITACHI LTD**(72) Inventor: **MATSUMOTO HIDEKAZU
NAKAMIKAWA TETSUAKI****(54) DATA PROCESSOR FOR EXECUTING
INSTRUCTION WITH GUARD****(57) Abstract:**

PURPOSE: To decide the guard conditions at high speed by performing comparison between the result obtained by a decoding means which decodes the executing conditions given by a instruction and the state of the data to be decided by said executing conditions and held by a state holding means.

CONSTITUTION: A means 140 decodes the guard conditions included in an instruction and a zero flag is set at every register stored in a register file 170 to decide whether the value of each register is equal to zero or not. A guard condition deciding circuit 150 compares the value of the most significant bit MSB of each register with the guard conditions which are designated by an instruction as the elements of the conditions to be decided. Then the guard conditions designated by an instruction are immediately decided. Thus the guard conditions are decided at high speed.

COPYRIGHT: (C)1989,JPO&Japio



⑫ 公開特許公報(A)

昭64-76325

⑬ Int. Cl.⁴
G 06 F 9/38

識別記号
330

庁内整理番号
J-7361-5B

⑭ 公開 昭和64年(1989)3月22日

審査請求 未請求 発明の数 1 (全8頁)

⑮ 発明の名称 ガード付命令を実行するデータ処理装置

⑯ 特 願 昭62-234568

⑰ 出 願 昭62(1987)9月18日

⑱ 発 明 者 松 本 秀 和 茨城県日立市久慈町4026番地 株式会社日立製作所日立研究所内

⑲ 発 明 者 中 三 川 哲 明 茨城県日立市久慈町4026番地 株式会社日立製作所日立研究所内

⑳ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

㉑ 代 理 人 弁理士 鶴 沼 辰 之 外1名

明 細 書

1. 発明の名称

ガード付命令を実行するデータ処理装置

2. 特許請求の範囲

1. 実行できるための条件を付加したガード付命令を含む命令とデータとを記憶するメモリと、前記メモリから命令を取り出し実行するプロセッサユニットとからなりガード付命令を実行するデータ処理装置において、

前記プロセッサユニットが、

命令で与えられる実行条件をデコードする手段と、

実行条件の判定対象となるデータの状態を生成する手段と、

生成されたデータの状態を保持する手段と、

前記デコード手段で得られた結果と前記状態保持手段に保持された状態とを比較判定し前記プロセッサユニットがガード付命令を実行するか否かを制御する手段と、

を備えたことを特徴とするガード付命令を実行

するデータ処理装置。

2. 特許請求の範囲第1項において、

前記実行条件デコード手段が、

実行条件の判定対象となるデータを保持している前記状態保持手段内のレジスタアドレスを命令から抽出する手段と、

前記比較判定条件を命令から抽出する手段とを含むことを特徴とするガード付命令を実行するデータ処理装置。

3. 特許請求の範囲第1項または第2項において、

前記判定対象データ状態生成手段が、

前記データ状態保持手段に書き込まれるデータがゼロであることを検出する手段からなり、

前記データ状態保持手段が、

前記ゼロ検出手段で検出された状態を前記データが書き込まれるレジスタ内に保持する手段からなることを特徴とするガード付命令を実行するデータ処理装置。

3. 発明の詳細な説明

[産業上の利用分野]

本発明は、データ処理装置に係り、特に、プログラムの実行を命令語自身で制御する機能を命令に持たせたマイクロプロセッサシステムに関する。

〔従来の技術〕

パイプライン処理は、複数の命令を同時に処理でき、計算機の高速化には有効な手段である。このため、最近ではマイクロプロセッサでもこの技術が取り入れられ、多段のパイプラインを採用する傾向にある。パイプラインの段数を増やすと、1サイクルに処理する仕事を少なくできるので、同一のプロセス技術でもマイクロプロセッサのマシンサイクル時間を短縮可能である。しかし、パイプライン段数が多くなると、逆に、分岐命令の処理が遅くなる。分岐命令などが実行される場合、それ以降の命令の処理をクリアするために、パイプラインがクリアされ、その結果、パイプラインの中に空きサイクルが生じてしまうからである。そこで、多段パイプラインを採るプロセッサでは、分岐処理におけるパイプラインの空きサイクルを削減することが重要である。

on Computer Architecture, 1986) に論じられている。

〔発明が解決しようとする問題点〕

上記従来技術は、ガード条件として命令で指定されたレジスタの値がゼロであるか否かを判定するには、演算を行わなければならないから、その判定に時間がかかる問題があった。また、レジスタの値がゼロまたはゼロでないというガード条件のみを用いていたので、それ以外の条件で、delayed slotに命令を埋め込むことができない欠点があった。

本発明の目的は、ガード条件の判定を高速に実行するデータ処理装置を提供することである。

〔問題点を解決するための手段〕

上記目的は、命令中のガード条件をデコードする手段を設け、各レジスタごとにそのレジスタの値がゼロであるか否かを示すゼロフラグを設け、ゼロフラグの値と各レジスタの値の正、負を判定するために各レジスタのMSB (Most Significant Bit) の値とを被判定条件の要素として命令で指

分岐処理により生じるパイプラインの空きサイクルを埋める手段として、delayed branchが採られる。この手法は、分岐命令実行後に生じる空きサイクル (delayed slotという) を活用するために、delayed slotに分岐命令の実行と無関係に実行される命令を置く方法である。delayed slot内の命令は、分岐処理に係わりなしに実行され、分岐処理で空いたパイプラインを埋める効果がある。ただし、delayed slotに置かれる命令は、分岐の成否に係わらず実行されるから、例えば条件分岐命令において条件不成立の場合に、delayed slotの命令を実行したくないときは、このslotに置くことはできない。その結果、delayed slotへの命令埋め込み率が低くなることがある。このslotの命令埋め込み率を向上させる1つの方法として、命令語中にその命令が実行できるための条件を付加したガード付命令が提案されている。この方法は、第13回コンピュータアーキテクチャ学会 (1986年) 第386ページから第395ページ (Proceeding of the 13th ACM Symposium

定されるガード条件と比較し命令で指定されたガード条件を直ちに判定する機構を設けることにより達成される。

〔作用〕

命令によつて指定された処理を実行後、その値をレジスタに書き込む場合、その値がゼロであれば、当該レジスタのゼロを示すゼロフラグを真とする。ガード条件の判定機構は、このフラグを見るだけでガード条件を判定できるので、ガード条件の判定が高速になる。

〔実施例〕

以下、図面を参照して、本発明の一実施例を詳細に説明する。

第2図は、本発明によるデータ処理装置の一実施例の全体構成を示している。データ処理装置は、プロセッサユニット (PU) 10と、メモリコントロールユニット (MCU) 20と、メインメモリ (MM) 30と、I/Oプロセッサ (IOP) 40とを主な構成要素とする。上記以外の要素も存在するが、本発明の理解には必要ないので、省略す

る。PU10は、MM30に格納されている命令およびオペランドデータをMCU20を介して取り込み、実行する。IOP40は、入出力制御用のプロセサである。本発明は、この中でPU10に関するものであるから、以下、PU10を主として説明する。もちろん、IOP40も基本的にはプロセサであるため、本発明が適用できる。信号線群15はPU10とMCU20、信号線群25はMCU20とMM30、信号線群35はMCU20とIOP40を各々結ぶものである。

第3図は、本実施例で採用している命令の形式を示す図である。本実施例では、RISC (Reduced Instruction Set Computer) アーキテクチャの命令形式を採る。すなわち、命令語を32ビット固定長にして、メモリを直接アクセスするのはロード、ストア命令だけに限定し、機能の複雑な命令はもたないことを基本とするアーキテクチャに従っている。第3図には、5種類の命令形式を示してある。Type-1は、レジスタ間の演算命令の形式であり、命令の処理内容を与える

オペレーションコードフィールド(OP)と、2組のソースオペランドとなるレジスタを指定するフィールド(Rs1, Rs2)と、結果が格納されるレジスタを指定するフィールド(Rd)と、ガード条件を指定するフィールド(GC)と、ガード条件として判定されるレジスタを指定するフィールド(Rg)とからなる。Type-2は、イミディエイトデータとの演算を行う命令形式であり、16ビットのイミディエイトデータを指定するフィールド(Imm16)をもつ。Type-3は、ロード命令またはストア命令の形式であり、ソース(またはデスティネーション)オペランドレジスタを指定するフィールド(Rd)と、アドレス計算のベースを与えるレジスタを指定するフィールド(Rb)と、ベースからのオフセットを与える16ビットのフィールド(Disp16)とからなる。Type-4は、ガード条件をもつロード命令(またはストア命令)の形式であり、8ビットのオフセットを与えるフィールド(Disp8)を含む。Type-5は、分岐命令の形式である。

ガード条件として、命令指定できるのは8種類である。その条件と命令でのビットパターンを、以下に示す。

1) always true	0 0 0
2) always false	0 0 1
3) equal zero	0 1 0
4) not equal zero	0 1 1
5) greater than zero	1 0 0
6) greater than zero or equal	1 0 1
7) less than zero	1 1 0
8) less than zero or equal	1 1 1

第1図は、本発明のPU10の内部構成を示すブロック図である。PU10は、命令フェッチポイント(FP)110と、命令パツファ(IB)120と、メモリアクセス制御回路(MAC)130と、命令デコーダ(DEC)140と、ガード条件判定回路(GCD)150と、ステージ制御回路(STC)160と、レジスタファイル(RF)170と、演算器(ALU)180と、ゼロ検出回路(ZD)190と、プログラムカウンタ(PC)

200と、メモリアドレスレジスタ(MAR)210と、メモリライトレジスタ(MWR)220と、リードデータレジスタ(RDR)230と、フラグ(F)240と、ラッチ(L)260、270と、セレクト(SEL)290と、ソースデータバス(ABUS, BBUS)410、420と、デスティネーションバス(CBUS)430とを主な構成要素とする。その他にも種々の構成要素が含まれるが、本発明の理解には必要ないので、省略する。

命令は、命令フェッチ(IF)、デコード(D)、実行(E)の3段のパイプライン処理により実行される。各パイプラインステージの大まかな処理を、以下に説明する。命令フェッチステージにおいて、命令をメモリ(MM30)からプリフェッチして、IB120に格納する。デコードステージにおいては、IB120から命令を取り出してデコードするとともに、そのデコード結果に基づいてレジスタファイル170の値を読み出す。また、ガード条件が指定されていれば、その条件を

判定して、結果をフラグ240にセットする。実行ステージにおいては、演算命令では演算の実行およびレジスタへの書き込み、ロード命令ではオペランドのアドレス計算後にメモリをアクセスしてオペランドをフェッチしてレジスタに書き込み、ストア命令ではオペランドのアドレス計算後にオペランドデータをメモリに書き込む。この3つのパイプラインステージは並列に実行され、3つの命令がパイプライン上で同時に処理される。

第4図は、命令デコーダ(DEC)140のさらに詳しい構成を説明するブロック図である。DEC140は、2組のデコーダ(DECF, DECS)141, 144と、ラッチ(L)142, 143と、ステージカウンタ(SC)145とを主な構成要素とする。DEC140はパイプライン処理のうち、デコードステージと実行ステージを制御する。デコードステージにおいては、IB120にある命令が、信号線120aを介して、DECF141に入力される。DECF141では、命令をデコードしてイミディエイトデータが命令に含まれる場合

は、それを取り出し信号線141aに出力し、ガード条件が含まれる場合は、その判定条件を信号線141bに、判定対象のレジスタアドレスを信号線141cに出力し、実行ステージで書き込みまたは読み出されるレジスタのアドレスを命令から取り出してラッチ142にセットし、実行ステージで必要となる制御情報をラッチ143にセットする。ラッチ142にラッチされたレジスタアドレスは、信号線142aを介してRF170に送られる。DECS144では、ラッチ143にセットされた情報から実行ステージの制御信号を出力する。実行ステージの処理が複数のサイクルにわたって行われる場合は、SC145を用いてステージの管理がなされる。

第5図は、RF170の内部構成を説明する図である。RF170は、32ビットレジスタを32本(R0~R31)内蔵する。各レジスタは、その値がゼロであるときに真となるゼロフラグ(Z)を備える。したがって、各レジスタのビット数は、実際には33ビットとなる。ビット0が

MSB (Most Significant Bit)、ビット31がLSB (Least Significant Bit) である。全てのレジスタのゼロフラグおよびMSB (ビット0) をまとめて信号線170a, 170bに出力する。その他、通常のアクセスをするためのリード/ライトバスを含むが、本発明の理解には必要ないので、省略する。

第6図は、ガード条件判定回路(GCD)150の内部構成を示すブロック図である。GCD150は、2組のセレクト(SEL)151, 152と、比較器(COMP)153とから構成される。SEL151とSEL152は、命令で指定されるガードレジスタのゼロフラグおよびMSBをセレクトし、COMP153に供給する。COMP153では、これらセレクトの出力と信号線141bに出力されたガード判定条件とから、ガード条件を判定し、ガード条件が成立しない場合は、信号線150aに“0”を、成立時には“1”を出力する。COMP153の入出力の関係を、第7図に示す。第7図では、信号線151aの値をZ、信号線152aの値を

Nとして表わしている。例えば、ガード判定の条件を示す信号線141bが“equal zero”の条件であると、Zを見て真であれば“1”を信号線151aに出力し、偽であれば“0”を出力する。信号線150aの値はフラグ240にセットされて、次のサイクルの実行ステージの動作を制御する。すなわち、フラグ240の出力は、信号線240aを介してステージ制御回路(STC)160に入力され、実行ステージの制御が行われる。STC160は、フラグ240に“1”がセットされた次のサイクルで実行ステージの動作を中止させる。これにより、ガード条件が不成立になると、そのガード条件を指定した命令は実行されないように動作する。

次に、各パイプラインステージでの処理を、第1図、第4図、第5図、第6図を使って詳細に説明する。

命令フェッチステージでは、フェッチする命令のアドレスをFP110に保持し、この内容をSEL290を通してMM30に送出し、対応するメモ

リアドレス上の命令をフェッチして、IB120に取り込む。MAC130は、このメモリアクセスに際して、メモリアクセスの起動信号の送出やメモリからの応答の受け取りなどを制御する。ただし、実行ステージがメモリをアクセス中である場合には、アクセスを待たされる。この優先制御は、MAC130でなされる。

デコードステージでは、IB120に保持されている命令を取り出してデコードし、そのデコード結果に基づいて、命令で指定されるレジスタの値をRF170から読み出し、ラッチ260またはラッチ270にセットする。イミディエイト値を含む命令では、その値を命令から抽出し、信号線141a, ABUS410を介して、ラッチ270にセットする。デスティネーションとなるレジスタがある場合、またはストア命令では、そのレジスタアドレスを命令から抽出してラッチ142にセットする。分岐命令の場合は、分岐アドレス計算のためのソースデータをラッチ260, 270にセットする。例えば、プログラムカウンタ相対

のアドレッシングでは、ラッチ260にPC200の値が、BBUS420, ABUS410を介して、ラッチ270に命令から抽出されたディスプレースメントが、それぞれセットされる。ガード条件が指定されていれば、その条件で指定されるレジスタの値と条件をGCD150に入力して判定し、結果をフラグ240にセットする。デコードステージでの制御は、DECF141からの制御信号線141dへの信号によつて行われる。

実行ステージでは、演算命令の場合、デコードステージで読み出したレジスタの値またはイミディエイト値がすでにラッチ260, 270にセットされているので、その値を使つてALU180で演算し、ラッチ142で指定されるRF170上のデスティネーションレジスタに結果を格納する。その際、ZD190によつてCBUS430上の値がゼロであるか否かを検出し、ゼロである場合には、デスティネーションレジスタのゼロフラグをセットする。

ロード命令の場合、ALU180の出力としてまずオ

ペランドのアドレスが得られるので、ALU180の出力をCBUS430を介してMAR210にセット後、SEL290を通してMM30にアドレスを出力し、オペランドをメモリから取り出す。取り出されたオペランドデータは、RDR230にラッチされたのち、ラッチ270を通してALU180に供給される。

ストア命令の場合、ALU180の出力として、まずオペランドのアドレスが得られるので、ALU180の出力をCBUS430を介してMAR210にセットする。次のサイクルでは、ラッチ142が指示するレジスタの内容をRF170から読み出して、ALU180を通してMVR220にセットするとともに、SEL290を通してMM30にアドレスを出力し、メモリ書き込みの起動信号をMM30に送出して、メモリに書き込む。

分岐命令の場合、ALU180の出力として分岐アドレスが得られるので、この値をCBUS430を介してPC200およびFP110にセットする。このとき、デコードステージで実行されている命令はキャンセルされない。ガード条件はデコードス

テージで評価され、その結果がフラグ240にセットされる。そこで、フラグ240が“0”にセットされた場合は、ガード条件が不成立であるため、STC160の制御により、実行ステージの動作が中止される。

〔発明の効果〕

本発明によれば、ガード条件の判定を演算によらずに高速実行でき、ガード条件の判定に伴うロス時間を削減可能である。また、分岐の実行／不実行に関わりなく、delayed branchのdelayed slotに命令を埋め込むことができるので、分岐命令におけるパイプラインの空サイクルが少なくなる。

4. 図面の簡単な説明

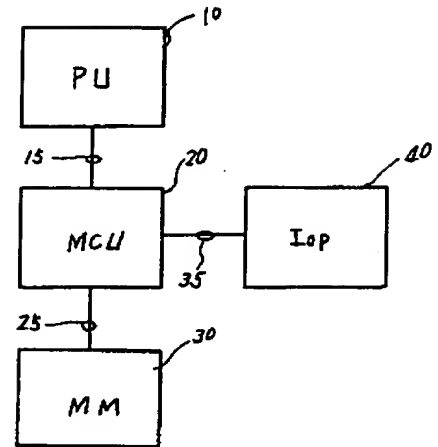
第1図は本発明によるデータ処理装置の一実施例の特徴的部分の構成を示すブロック図、第2図は本発明によるデータ処理装置の全体構成の一例を示すブロック図、第3図は本発明の実施例で採用している命令形式を示す図、第4図、第5図、第6図は第1図特徴的部分のさらに詳細な構成を

示すブロック図、第 7 図は第 6 図回路の判定例を示す図である。

110…命令フェッチポイント、120…命令バスファ、130…メモリアクセス制御回路、140…命令デコーダ、141, 144…デコーダ、142, 143…ラッチ、145…ステージカウンタ、150…ガード条件判定回路、151, 152…セクタ、153…比較器、160…ステージ制御回路、170…レジスタファイル、180…演算器、190…ゼロ検出回路、200…プログラムカウンタ、210…メモリアドレスレジスタ、220…メモリライトレジスタ、230…リードデータレジスタ、240…フラグ、260, 270…ラッチ、290…セクタ、410, 420…ソースデータバス、440…デスティネーションバス。

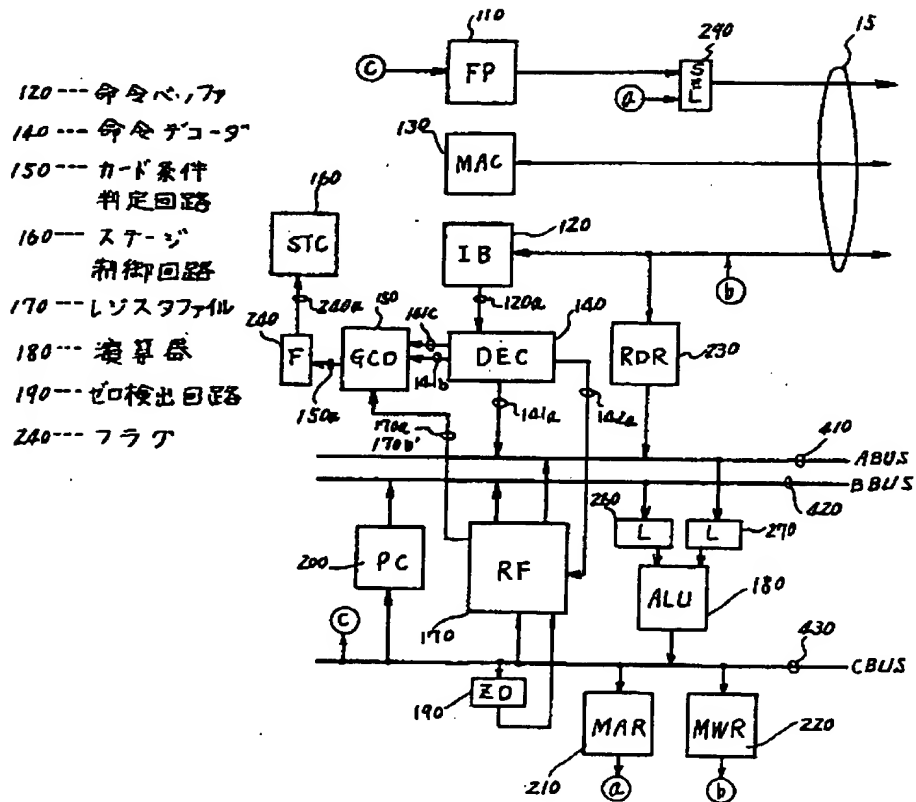
代理人 弁理士 橋沼辰之

第 2 図

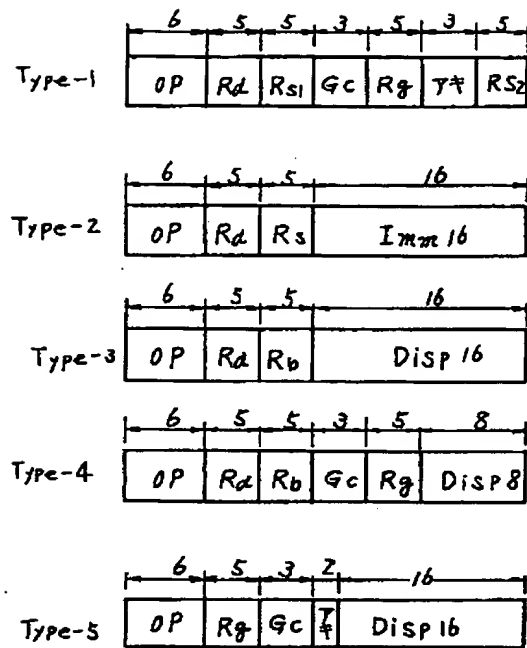


10…プロセサユニット
20…メモリコントローラ
30…メインメモリ
40…入出力プロセサ

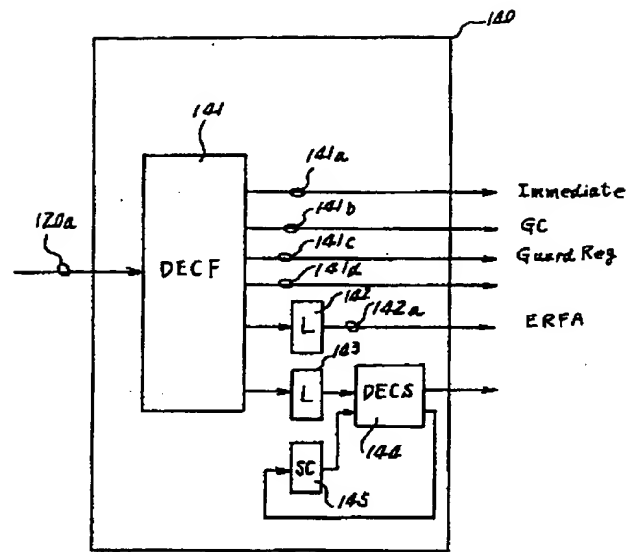
第 1 図



第 3 図



第 4 図

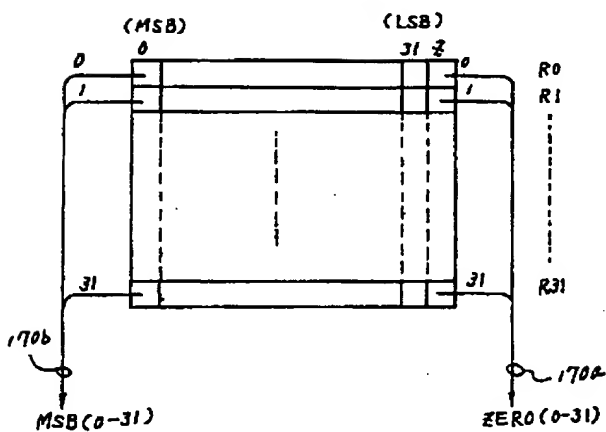


141, 144 --- デコーダ

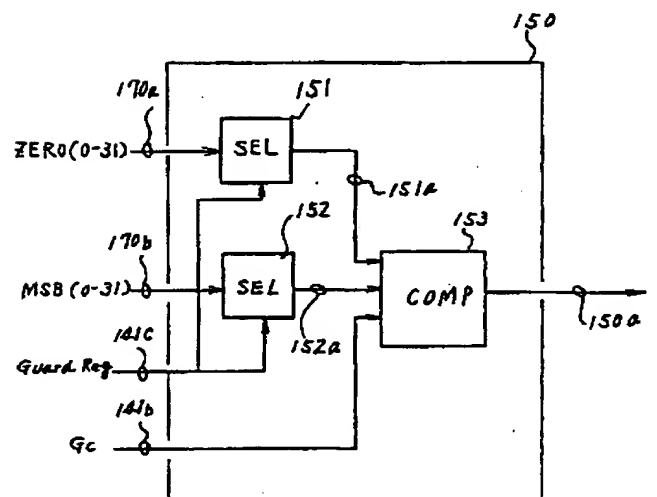
142, 143 --- ラッチ

145 --- ステジカウンタ

第 5 図



第 6 図



151, 152 --- セレクタ

153 --- 比較器

第 7 図

信号線 141b	出力信号線 150a
000	1
001	0
010	\bar{Z}
011	\bar{Z}
100	$\bar{N} \wedge \bar{Z}$
101	$\bar{N} \vee Z$
110	$N \wedge \bar{Z}$
111	$N \vee Z$